

## **REMARKS**

Initially, Applicant expresses appreciation to the Examiner for the courtesies extended during the recent in-person interview held with Applicant's representatives. The amendments and arguments presented herein are consistent with the discussions of the interview. Accordingly, reconsideration for the above-identified application is now respectfully requested.

The Office Action, mailed September 8, 2006, considered and rejected claims 1-46. Claims 11-19 and 29-46 were rejected under 35 U.S.C. § 101 as being drawn to non-statutory subject matter. Claims 36-46 were also rejected under 35 U.S.C. § 112, second paragraph as being indefinite.<sup>1</sup> Claims 1-7, 9-16, 28-33, 35-42, 45 and 46 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Barnes (U.S. Patent No. 7,096,419) in view of Chinnici (U.S. Publ. No. 2003/0191803). Claims 8, 17, 27, 34, 43 and 44 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Barnes (U.S. Patent No. 7,096,419) in view of Chinnici (U.S. Publ. No. 2003/0191803), and further in view of Ardoin (U.S. Patent No. 6,052,691).<sup>2</sup>

By this paper, claims 1, 5, 8, 11, 15, 17, 20, 29, 34 and 36 have been amended, claims 43 and 44 cancelled, and no claims added. Accordingly, following this paper, claims 1-42, 45 and 46 remain pending, of which claims 1, 11, 20, 29 and 36 remain pending.

As discussed during the interview, Applicant's invention generally relates to serializing user interface objects of custom object types and serialization formats. As reflected in claim 1, for example, an exemplary method for serializing one or more objects from an initial representation to at least one subsequent representation includes providing a serialization manager that coordinates one or more standard serialization providers identifying standard serializers, and which loads, as necessary, one or more custom serialization providers that identify custom serializers for custom object types or serialization formats. A request is then made to the serialization manager for a serializer for an object graph that comprises an object of a particular type and for a particular serialization format. Such a request is made as part of a cut, copy or paste operation and, due to the requested operation, the serializer produces a snippet of

---

<sup>1</sup> In light of the claim amendments above, Applicant submits that the rejections under 35 U.S.C. §§ 101 and 112 are now moot.

<sup>2</sup> Although the prior art status of the cited art is not being challenged at this time, Applicant reserves the right to challenge the prior art status of the cited art at any appropriate time, should the need arise. Accordingly, any arguments and amendments made herein should not be construed as acquiescing to any prior art status of the cited art.

code sufficient to undo or redo a change to the object graph. The serializer is then called to serialize the object graph.

Claims 11, 20, 29 and 36 are directed to methods and computer program products and generally correspond to the method of claim 1.

As discussed during the interview, and as noted in the Office Action, the Barnes and Chinnici references, which generally relate to object serialization and source code generation, do not disclose or suggest serializing an object in response to a cut, copy or paste operation or the production of a snippet of code by the serializer sufficient to undo or redo a change to the object graph.

Applicant also respectfully submits that the Ardoin reference is deficient in this regard, as noted in the interview. In particular, Ardoin generally relates to CAD/CAM software systems for managing objects so that when an object is modified, copied, or deleted, referential integrity is maintained, and so as to allow for undoing or reversing actions taken on the object. (Col. 1, ll. 32-43; Col. 3, ll. 13-29). More particularly, Ardoin describes a system which includes a relations subsystem that describes the associative and constrain relationships between various objects by using an associative graph. (Col. 6, ll. 28-35). Within the associative graph are nodes that are connected and represent related entities, and edges between the nodes, which represent the relationships between the entities. (Col. 6, ll. 28-44). The graph itself may be modified when nodes/entities are modified, copied, or deleted. (Col. 6, ll. 35-39). Further, the system of Ardoin can include an optional undo interface. When objects are modified, the undo interface writes modifications to a log, which log can be read to restore a value. (Col. 41, ll. 47-54).

While Ardoin generally describes modifying, copying, and deleting objects within a graph, Applicant respectfully submits that Ardoin fails to teach or suggest Applicant's invention as claimed. In particular, Ardoin does not teach or suggest that the cut, copy, or past operation causes a serializer to be requested, or that the requested serializer produces a snippet of code, as recited in combination with the other claim elements. In fact, while Ardoin appears to disclose modifying relationships between nodes in response to a modification of a node, it appears that Ardoin fails to include any mention of serializing the associative graph or even changing the format of the graph. Moreover, Ardoin discloses that a log file, rather than a snippet of code, is generated to facilitate an undo operation.

In view of the foregoing, and for the other reasons discussed during the interview, Applicant respectfully submits that the other rejections to the claims are now moot and do not, therefore, need to be addressed individually at this time.<sup>3</sup> It will be appreciated, however, that this should not be construed as Applicant acquiescing to any of the purported teachings or assertions made in the last action regarding the cited art or the pending application, including any official notice. Instead, Applicant reserves the right to challenge any of the purported teachings or assertions made in the last action at any appropriate time in the future, should the need arise. Furthermore, to the extent that the Examiner has relied on any Official Notice, explicitly or implicitly, Applicant specifically requests that the Examiner provide references supporting the teachings officially noticed, as well as the required motivation or suggestion to combine the relied upon notice with the other art of record.

In the event that the Examiner finds remaining impediment to a prompt allowance of this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney.

Dated this 8<sup>th</sup> day of December, 2006.

Respectfully submitted,



RICK D. NYDEGGER  
Registration No. 28,651  
JENS C. JENKINS  
Registration No. 44,803  
COLBY C. NUTTALL  
Registration No. 58,146  
Attorneys for Applicant  
Customer No. 047973

JCJ:CCN:dms  
CRB0000003059V001

---

<sup>3</sup> With respect to claim 8, however, and as discussed during the interview, Applicant also notes that it appears that Barnes and the other cited art fail to disclose wherein the snippet of code is produced without producing a class representation of the object class. For example, Barnes discloses a method in which Java objects are generated and in which the class of such objects is determined. (Col. 5, ll. 17-24, 64-67). Similarly, in Chinnici, an extensible type mapping system expressly includes mapping between and the generation of class objects (¶¶ 128, 129 and 137-140).